Design of a Reinforcement Learning PID controller

1st Zhe Guan
KOBELCO Construction Machinery,
Dream-Driven Co-Creation Research Center
Hiroshima University
Higashi-Hiroshima, Japan
guanzhe@hiroshima-u.ac.jp

2nd Toru Yamamoto
Graduate School of Advanced Science and Engineering
Hiroshima University
Higashi-Hiroshima, Japan
yama@hiroshima-u.ac.jp

Abstract—**This paper addresses a design problem of a Proportional-Integral-Derivative (PID) controller with new adaptive updating rule based on Reinforcement Learning (RL) approach for nonlinear systems. A new design scheme that RL can be used to complement the conventional control technology PID is presented. In this study, a single Radial Basis Function (RBF) network is introduced to calculate the control policy function of Actor and the value function of Critic simultaneously. Regarding to the PID controller structure, the inputs of RBF network are system error, the difference of output as well as the second order difference of output, and they are defined as system states. The Temporal Difference (TD) error in this study is newly defined and involves the error criterion which is defined by the difference between one-step ahead prediction and the reference value. The gradient descent method is adopted based on TD error performance index, then the updating rules can be obtained. Therefore, the network weights and the kernel function can be calculated in an adaptive manner. Finally, the numerical simulations are conducted in nonlinear systems to illustrate the efficiency and robustness of the proposed scheme.**

Index Terms—**Adaptive control, PID control, Reinforcement Learning**

I. Introduction

PID control is considered as an effective tool and is one of the most common control schemes and has been dominated the majority of industrial processes and mechanical systems, since it is of versatility, high reliability and ease of operation [1]. PID controllers can be manually tuned appropriately by the operators and control engineers based on the empirical knowledge when the mathematical model of the controlled plant is unknown. Some classical tuning methods, such as Ziegler-Nichols method [2] and Chien-Hrones-Reswich method [3], are applied to the process control and the performance then is significantly outperformed compared to the one that is manually tuned. However, those methods work well for simple controlled plants, but for complex systems with non-linearity, the performance can not be guaranteed due to the presence of uncertainty and unknown dynamics. In addition, there is not existing an exact model which can be built from the real systems. Therefore, the adaptive PID control has been received considerable attentions in last 20 years in order to deal with those systems.

Several adaptive PID control strategies which include model-based adaptive PID control in [5], [6], [7], adaptive PID control based on neural network [8], [9]. It has been clarified that model-based adaptive PID control needs an assumption that the established model could represent the true plant dynamics exactly [10]. However, modeling complex systems are time-consuming and lack of accuracy, hence the PID parameters may not be adjusted in a proper way. On the other hand, the adaptive PID control based on neural network adopts the supervised learning to optimize the network parameters. Therefore, there are some limitations in the application of those methods, such as the teaching signal is hard to be obtained, and it is difficult to predict values for unlabeled data. As a result, the adaptive PID control based on various more advanced machine learning technologies has been discussed with the rapid development of computer science.

Machine learning technology has been increasingly applied in various fields, including the control engineering community introduced in the [11]. The numerous algorithms have been developed to achieve desirable performance and intelligent decision making for many complex control problems. On the other hand, the great advances in computing power have enabled us to implement the sophisticated learning algorithms in practice. Bishop *et al.* [12] has clarified that machine learning is customarily divided into three classes of algorithms: supervised learning, unsupervised learning and reinforcement learning. Reinforcement learning (RL) differs significantly from both supervised and unsupervised learning [13]. A definition of RL from [14] is expressed as: a RL agent has the goal of learning the best way to accomplish a task through repeated interactions with its environment. It already enable innovations in broad applications [15], [16]. From the control perspective, RL refers to an agent (controller) that interacts with its environment (controlled system) and then modifies its actions (control signal) [17]. It has strong potential to combine the RL technology with the adaptive PID control to have an impact on process control applications, and it has been investigated in studies [4], [18], [19], [20], [21]. In the literature [4], the reinforcement signal was defined by a error between current output and reference signal, which may cause the prediction loss. [18], [20], [21] adopted the same updating rule and did not provide the trajectories of PID parameters. Moreover the updating rule for three parameters is compacted in one equation. The model based design method was given in [19].

Based on the observations above, this paper considers a

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

PID controller with new adaptive updating rule based on RL technology for nonlinear systems. Actor-Critic structure [22] is one of classes in RL technology and is regarded as an benchmark in some design method, in which an actor component applies control signal to a system, and a critic component assesses the value of the output simultaneously. Besides, it has been investigated that the Actor-Critic structure is the most general and successful to date [13]. In this study, the idea of realization of an actor and a critic by using RBF network where it can reduce demands of storage and avoid repetitive calculation. Under the Actor-Critic structure based on RBF network, the new adaptive updating rule can be designed.

The main contributions of this study are summarized as follows. First, the reinforcement signal is re-defined by considering the one-step predictive output, therefore, the prediction error is involved in the TD error. Second, the new adaptive updating rule can be calculated based on the one-step TD error. Finally, the proposed scheme is model free design, which is much suitable for complex real systems.

The remainder of this paper is organized as follows. The problem formulation is discussed in Section 2, where two reasonable assumptions are introduced as well. In Section 3, the adaptive PID controller based on Actor-Critic algorithm is proposed. Numerical simulation and comparative study are provided to illustrate the efficiency and feasibility in Section 4. Finally, Section 5 concludes this paper.

II. Problem Statement

Consider the following discrete-time systems described by nonlinear dynamics in the affine state space difference equation form

$$\begin{aligned} x(t+1) =& f(x(t)) + g(x(t))u(t) \\ y(t) =& h(x(t), u(t-1)), \end{aligned} \tag{1}$$

with state $x(\cdot) \in R^m$, control input $u(\cdot) \in R^n$ and output $y(\cdot)$. Since in the RL technology, the detail information of a model can be unknown, therefore, the above system can be generalized to a compact form:

$$\begin{aligned} x(t+1) =& F(x(t), u(t)) \\ y(t) =& h(x(t), u(t-1)). \end{aligned} \tag{2}$$

It is required to provide two assumptions on the above system in order to capture the ideas about RL technology.

Assumption 2.1: The above system satisfies the 1-step Markov property since the state at time $t+1$ only depends on the state and inputs at the previous time t, independent with the historical data.

This assumption is under the framework of Markov decision processes (MDP), whose objective is to achieve a specified goal through a satisfactory control policy. It is defined in a similar way with RL technology, which makes it significant impact in combining control problem with RL technology. MDP is a mathematically idealized form of the RL problem [14].

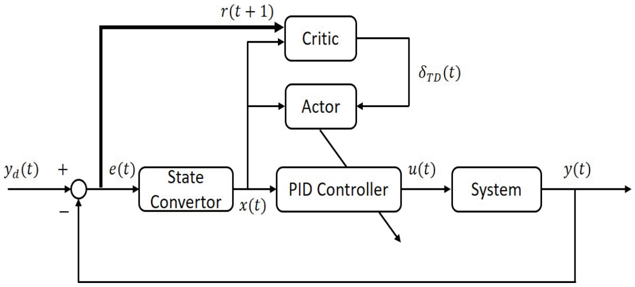

Fig. 1. The block diagram of the proposed scheme

Assumption 2.2: The sign of partial derivatives of $h(\cdot)$ with respect to all arguments is known, and it is also regarded as the sign of system Jacobian [25].

A. Controller structure

It is well recognized that a PID controller is applied to process systems, therefore, the derivative kick sometimes has an impact on the performance of the closed-loop system. As a consequence, this paper introduces the following velocity-type PID controller which can reduce the derivative kick:

$$u(t) = u(t-1) + K_I(t)e(t) - K_P(t)\Delta y(t) - K_D(t)\Delta^2 y(t), \tag{3}$$

that is

$$\Delta u(t) = \boldsymbol{K}(t)\boldsymbol{\Theta}(t), \tag{4}$$

where, $\boldsymbol{\Theta}(t)$ is defined as

$$\boldsymbol{\Theta}(t) := [e(t), -\Delta y(t), -\Delta^2 y(t)]^T, \tag{5}$$

and it is regarded as system state. Δ denotes the difference operator defined by $\Delta := 1 - z^{-1}$. The $\Delta^2 y(t)$ then becomes:

$$\Delta^2 y(t) = y(t) - 2y(t-1) + y(t-2) \tag{6}$$

$\boldsymbol{K}(t) := [K_I(t), K_P(t), K_D(t)]$ is a vector of control parameters. $e(t)$ is the control error and is defined by the difference between reference signal y_d and system output y as follows,

$$e(t) = y_d(t) - y(t). \tag{7}$$

B. Objective

The schematic diagram of the proposed method is show in Fig. 1, in which the system state $\boldsymbol{\Theta}(t)$ is constructed based on $e(t)$ and current system output firstly, and then they will be used as inputs to the Actor-Critic structure. The Actor tunes the controller on-line using the observed system state along the system trajectory, while the Critic, which receives the system state and reinforcement signal $r(t+1)$, assesses the performance and produces the Temporal Difference (TD) error. The TD error is viewed as a crucial basis for updating the parameters. As a result, the objective of this paper is to design a PID controller with new adaptive updating rule under the Actor-Critic structure.

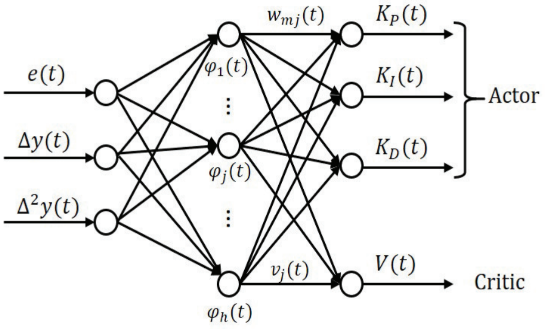

Fig. 2. RBF network topology with Actor-Critic structure

III. Adaptive Controller Design

The proposed algorithm will be explained in detail in this section.

A. Temporal Difference (TD) error

We will first introduce a value function which is defined as

$$V(t) = \sum_{i=t}^{\infty} \gamma^{i-t} r(x(i), u(i)) \tag{8}$$

with $0 < \gamma \leq 1$ a discount factor and $u(t)$ control signal. Function $r(x(i), u(i))$ is known as reinforcement signal, and can be selected based on quadratic function.

By rewriting (8) as

$$V(t) = r(x(t), u(t)) + \gamma \sum_{i=t+1}^{\infty} \gamma^{i-(t+1)} r(x(i), u(i)). \tag{9}$$

Instead of evaluating the infinite sum of above equation, one can use the current control signal $u(t)$ to solve the following difference equation equivalent:

$$V(t) = r(x(t), u(t)) + \gamma V(t+1), V(0) = 0. \tag{10}$$

This equation is also known as *Bellman equation*.

Based on the Bellman equation, a TD error can be defined as the difference between the two sides:

$$\delta_{TD}(t) = r(x(t), u(t)) + \gamma V(t+1) - V(t). \tag{11}$$

If the Bellman equation holds, the TD error is zero. Therefore, the current control signal may be regarded as the optimal control policy at each time t.

B. Actor-Critic learning based on RBF network

The RBF network has been used as a technique to identify parameters by performing function mappings. The simple structure, parameters convergence and adequate learning are recognized as merits of RBF network and are discussed in [23]. As a consequence, the implementation of Actor-Critic is used by RBF network in this study, and the network topology is shown in Fig. 2. It consists of three-layer neural networks.

The input layer consists the available process measurements and system states are constructed. On the basis of the RBF network topology, it allows to pass the system states to the hidden layers which are shared by the Actor and the Critic directly. The control signal $u(t)$ and value function are generated by means of a simpler way that is the weighted sum of the function value associated with units in hidden layer [24]. The detail of each layer is described as follows.

The input layer includes the system state variable x_i where i is an input variable index. Input vector $\mathbf{\Theta}(t) \in R^3$ is passed to the hidden layer and is used to calculate the output of hidden unit.

In hidden layer, $\Phi_j(t)$ is a vector which contains the elements $[\phi_1(t), \cdots, \phi_h(t)]$, where h is the number of the hidden units. The Gaussian function is selected as a kernel function of the hidden unit of RBF network, therefore, the output $\Phi(t)$ is shown as following:

$$\Phi_j(t) = exp\left(-\frac{||\mathbf{\Theta}(t) - \boldsymbol{\mu}_j(t)||^2}{2\sigma_j^2(t)}\right), j = 1, 2, 3, \ldots, h \tag{12}$$

where, $\boldsymbol{\mu}_j$ and σ_j are the center vector and width scalar of the unit, respectively. The center vector is defined as follows.

$$\boldsymbol{\mu}_j(t) := [\mu_{1j}, \mu_{2j}, \mu_{3j}]^T.$$

The third layer is called output layer where the outputs of the Actor and the Critic are involved. It should be noted that as mentioned previously the outputs are calculated in a simple and direct way. Therefore, it can yield the PID parameters $\boldsymbol{K}(t)$ in the following:

$$K_{P,I,D}(t) = \sum_{j=1}^{h} w_j^{P,I,D}(t)\Phi_j(t), \tag{13}$$

with the weights w_{nj} between the jth hidden unit and output layer of the Actor. The value function of critic part can be obtained as follows:

$$V(t) = \sum_{j=1}^{h} v_j(t)\Phi_j(t), \tag{14}$$

where $v_j(t)$ denotes the weight between the jth hidden unit and output layer of the Critic.

Those various output weights can be trained by gradient-based learning algorithm. Therefore, we can obtain the adaptive updating rule under user-specified parameters. Recall the (5), the reinforcement signal in this study is defined as

$$r(x(t), u(t)) := \frac{1}{2}(y_d(t+1) - y(t+1))^2, \tag{15}$$

which indicates the difference between predictive performance and reference value. The TD error then becomes

$$\delta_{TD}(t) = \frac{1}{2}(y_d(t+1) - y(t+1))^2 + \gamma V(t+1) - V(t). \tag{16}$$

As a result, the cost function in this study is denoted in the following:

$$J(t) = \frac{1}{2}\delta_{TD}^2(t). \tag{17}$$

Thus, the partial differential equations with respect to each output weight of the Actor are developed as

$$w_j^P(t+1) = w_j^P(t) - \alpha_w \frac{\partial J(t)}{\partial w_j^P(t)} \tag{18}$$

where, α_w is a learning rate, and

$$\begin{aligned}\frac{\partial J(t)}{w_j^P(t)} =& \frac{\partial J(t)}{\partial \delta_{TD}(t)} \frac{\partial \delta_{TD}(t)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_P(t)} \frac{\partial K_P(t)}{\partial w_j^P(t)} \\ =& \delta_{TD}(y(t) - y(t-1))\Phi_j(t) \frac{\partial y(t+1)}{\partial u(t)}.\end{aligned} \tag{19}$$

$$\begin{aligned}\frac{\partial J(t)}{w_j^I(t)} =& \frac{\partial J(t)}{\partial \delta_{TD}(t)} \frac{\partial \delta_{TD}(t)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_I(t)} \frac{\partial K_I(t)}{\partial w_j^I(t)} \\ =& - \delta_{TD} e(t)\Phi_j(t) \frac{\partial y(t+1)}{\partial u(t)}.\end{aligned} \tag{20}$$

$$\begin{aligned}\frac{\partial J(t)}{w_j^D(t)} =& \frac{\partial J(t)}{\partial \delta_{TD}(t)} \frac{\partial \delta_{TD}(t)}{\partial y(t+1)} \frac{\partial y(t+1)}{\partial u(t)} \frac{\partial u(t)}{\partial K_D(t)} \frac{\partial K_D(t)}{\partial w_j^D(t)} \\ =& \delta_{TD}(y(t) - 2y(t-1) + y(t-2))\Phi_j(t) \frac{\partial y(t+1)}{\partial u(t)}.\end{aligned} \tag{21}$$

It should be noted that a *prior information* about the system Jacobian $\partial y(t+1)/\partial u(t)$ is required in order to calculate the above equations. Here, we consider a relation $\epsilon = |\epsilon|\text{sign}(\epsilon)$, therefore, the system Jacobian is obtained by the following equation.

$$\frac{\partial y(t+1)}{\partial u(t)} = \left| \frac{\partial y(t+1)}{\partial u(t)} \right| \text{sign}\left(\frac{\partial y(t+1)}{\partial u(t)} \right), \tag{22}$$

with $\text{sign}(\epsilon) = 1(\epsilon > 0), -1(\epsilon < 0)$. Based on the above assumption, the sign of the system Jacobian can be obtained. [25]. The updating rule for output weight of the Critic is

$$\begin{aligned}v_j(t+1) =& v_j(t) - \alpha_v \frac{\partial J(t)}{\partial v_j(t)} \\ =& v_j(t) + \alpha_v \delta_{TD}(t)\Phi_t(t),\end{aligned} \tag{23}$$

with a learning rate α_v.

The centers and the widths of hidden units in the hidden layer are considered to be updated in the following ways:

$$\begin{aligned}\mu_{ij}(t+1) =& \mu_{ij}(t) - \alpha_\mu \frac{\partial J(t)}{\partial \mu_{ij}(t)} \\ =& \mu_{ij} + \alpha_\mu \delta_{TD}(t) v_j(t)\Phi_j(t) \frac{\psi_i(t) - \mu_{ij}(t)}{\sigma_j^2(t)},\end{aligned} \tag{24}$$

while,

$$\begin{aligned}\sigma_j(t+1) =& \sigma_j(t) - \alpha_\sigma \frac{\partial J(t)}{\partial \sigma_j(t)} \\ =& \sigma_j + \alpha_\sigma \delta_{TD}(t) v_j(t)\Phi_j(t) \frac{||\psi_i(t) - \sigma_j(t)||^2}{\sigma_j^3(t)},\end{aligned} \tag{25}$$

where α_μ and α_σ are learning rates of center and width, respectively.

C. Algorithm summary

The every design step of the proposed adaptive PID controller under Actor-Critic structure based on RBF network is presented in *Algorithm* 1. To achieve a better performance, an explanation has to be clarified that the user-specified parameters are inevitable. Some limited trial and errors have to be conducted when the algorithm is implemented.

Algorithm 1 Adaptive PID controller under Actor-Critic based on RBF network

```
Initialize instant $t = 0$, control input signal $u(0)$ and reference signal $y_d(t)$.
Initialize the parameters $w_j^{P,I,D}(0)$, $v_j(0)$, $\mu_{ij}(0)$, $\sigma_j(0)$ and set the values for the use-specified learning rates $\alpha_w$, $\alpha_v$, $\alpha_\mu$, $\alpha_\sigma$.
for $t = 1 : EndTime$
    Measure the system output $y(t)$ and then the system error $e(t)$ can be obtained.
    Compute the kernel function (12) in hidden layer.
    Calculate the output of Actor, that is the current PID parameters from (4), and the output of Critic value function $V(t)$ from (14) at time $t$.
    Obtain the current control signal by
        $\Delta u(t) = K_I(t)e(t) - K_p(t)\Delta y(t) - K_d(t)\Delta^2 y(t).$
    Apply the control signal to controlled system and yield predictive value of system output $y(t+1)$.
    Construct the system state by the predictive value:
        $\mathbf{\Theta}(t+1) := [e(t+1), \Delta y(t+1), \Delta^2 y(t+1)]^T.$
    Calculate the value function $V(t+1)$ from (14).
    Obtain the TD error $\delta_{TD}(t)$ from (16).
    Update the weights of the PID parameters by (19) - (21) and the weights of the value function according to (23).
    Update the centers and the widths of RBF kernel functions by (24) - (25).
end for
```

IV. Numerical Simulations

The numerical simulation and comparative study are conducted in this section in order to evaluate the efficiency and feasibility of the proposed scheme. Consider the following non-linear system from [26]:

$$y(t+1) = \frac{y(t)y(t-1)[y(t)+2.5]}{1 + y(t)^2 + y(t-1)^2} + u(t) + \xi(t), \tag{26}$$

where $\xi(t)$ denotes the Gaussian noise with zero mean and variance of 0.01^2. It should be noted that the static property of this non-linear system is not provided because of page limitation.The reference signal values are set as follows:

$$y_d(t) = \begin{cases} 2.5(0 \leq t < 100) \\ 3.5(100 \leq t < 200) \\ 1(200 \leq t < 300) \\ 3(300 \leq t < 400) \end{cases}. \tag{27}$$

The user-specified learning rates included in the proposed are summarized as follows:

$$\alpha_w = 0.013, \alpha_v = 0.021, \alpha_\mu = 0.0025, \alpha_\sigma = 0.009,$$

and the coefficient γ equals to 0.98. The hidden units in topology RBF network are decided as 3. The initial PID parameters in the proposed scheme are set as

$$\boldsymbol{K}(0) = [0, 0, 0]^T.$$

There is no need to give the initial value in the proposed scheme.

The simulation results are presented in Fig. 3, where the output signal can track the reference signal by employing the proposed scheme. Regardless of the strong non-linearity, the proposed scheme can work well when the reference signal is changed. Moreover, the PID parameters are depicted in Fig. 4, where they can be updated based on the updated weights. Furthermore, they ultimately tended to reach constant values, which illustrates that the new updating rule works well within a certain range. The TD error is provided as well in Fig. 5, where the value is close to zero at steady state.

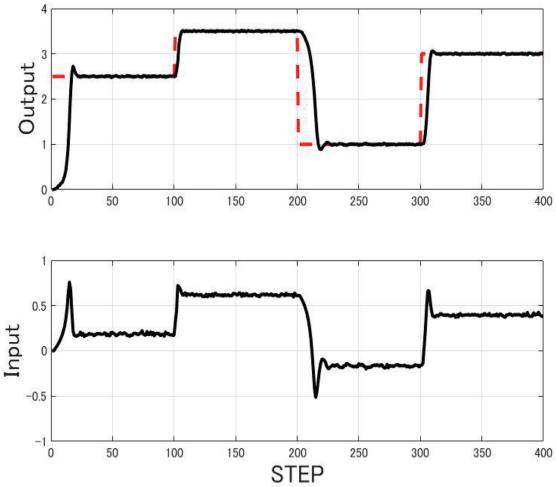

Fig. 3. Control result obtained by the proposed scheme

The comparative study for the proposed scheme is discussed by employing a conventional adaptive PID tuning method. The normal gradient method is adopted to update the PID parameters. The control results are shown in Fig. 6 and Fig. 7, respectively. Fig. 6 shows the practical tracking problem can be solved, however, the overshoot is apparent larger than that one from the proposed scheme. This should be due to the strong non-linearity in the system.

V. Conclusions

This paper has studied a novel adaptive PID controller under the Actor-Critic structure based on RBF network for nonlinear systems. A new adaptive updating rule was presented via weights update in the network. First, the conventional PID controller combined with the reinforcement learning on the basis of RBF network, and the PID tuned in an on-line manner. The reinforcement signal was defined by considering the predictive output, thus, the update could perform in an accurate way. Then, the hidden layer of RBF network was shared by the Actor and the Critic. The storage space could be saved and the computation cost was reduce for the outputs of the hidden units. In addition, the initial PID parameters are set as zero, which means there is no need to know the *prior* knowledge on the controlled system. Finally, numerical simulations were given to indicate the efficiency and feasibility of the proposed scheme for complex nonlinear systems. The PID parameters based on the new adaptive updating rule reached to constant values. The deficiency of the proposed scheme is that some user-specified parameters needed to set by empirical trials and they can not be exceeded a certain range. An interesting problem is to how one can set the initial

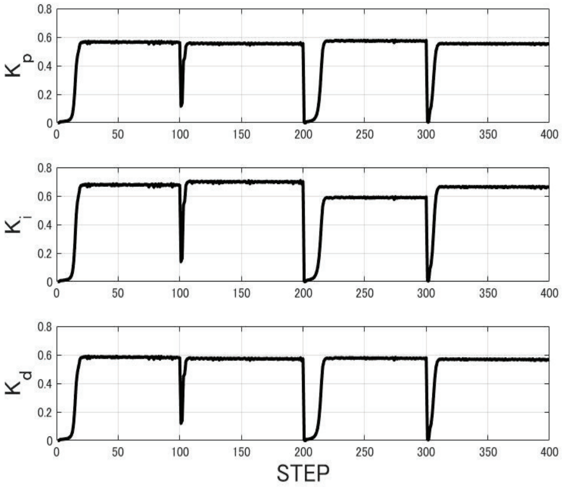

Fig. 4. Trajectories of adaptive PID parameters

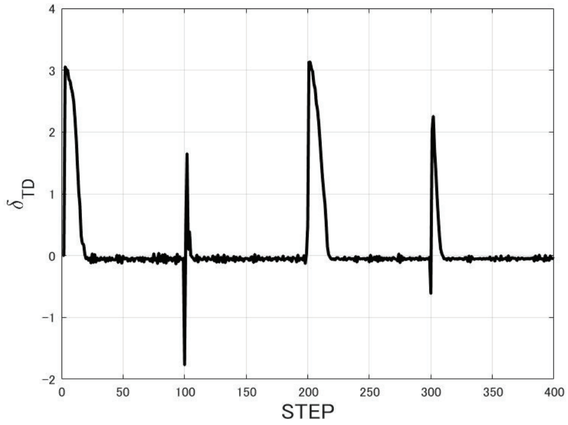

Fig. 5. Trajectories of TD error

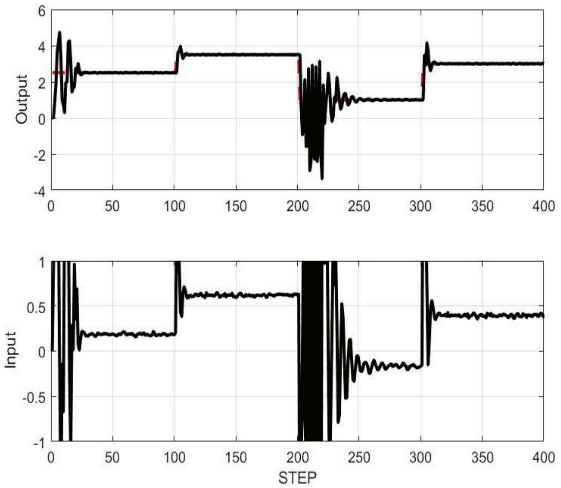

Fig. 6. Control result obtained by the conventional scheme

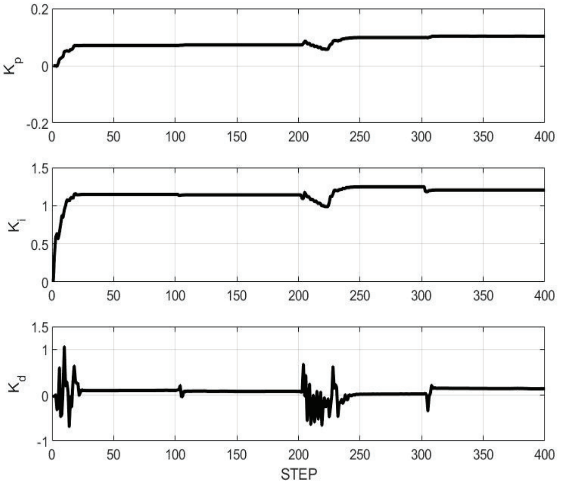

Fig. 7. Control result obtained by the conventional scheme

parameters properly. Furthermore, the proposed scheme will be employed in a real system to verify the effectiveness from the practical point of view.

References

[1] K. J. Åström and T. Hägglund. PID Controllers: Theory, Design and Tuning - 2nd edition. Instrument Society of America, 1995.

[2] J. G. Ziegler and N. B. Nichols. Optimum Settings for Automatic Controllers. Trans. of the ASME, Vol. 64, pp. 759-768, 1942.

[3] K. L. Chien, J. A. Hrones, and J. B. Reswick. On the automatic control of generalized passive systems. Trans. of the ASME, Vol. 74, No. 2, pp. 175-185, 1952.

[4] K. S. Hwang, S. W. Tan, and M. C. Tsai. Reinforcement Learning to Adaptive Control of Nonlinear Systems. IEEE Trans. on Systems, Man, and Cybernetics Part: B Cybernetics, Vol. 33, No. 3, pp. 514-521, 2003.

[5] W. D. Chang, R. C. Hwang and J. G. Hsieh. A multi-variable on-line adaptive PID controller using auto-tuning neurons. Engineering Application of Artificial Intelligence 16, pp. 57-63, 2003.

[6] T. Yamamoto and S. L. Shah. Design and Experimental Evaluation of a Multivariable Self-Tuning PID Controller. IEEE Proc. of Control Theory and Applications, Vol. 151, No. 5, pp. 645-652, 2004.

[7] D. L. Yu, T. K. Chang and D. W. Yu. A stable self-learning PID control for multi-variable time varying systems. Control Engineering Practice, Vol. 15, No. 12, pp. 1577-1587, 2007.

[8] J. H. Chen and T. C. Huang. Applying neural networks to on-line updated PID controllers for nonlinear process control. Journal of Process Control, Vol. 14, No. 2, pp. 211-230, 2004.

[9] Y. T. Liao, K. Koiwai and T. Yamamoto. Design and implementation of a hierarchical-clustering CMAC PID controller. Asian Journal of Control, Vol. 21, No. 3, pp. 1077-1087, 2019.

[10] Z. S. Hou, R. H. Chi and H. J. Gao. An overview of dynamic linearization based data-driven control and applications. IEEE Trans. on Industrial Electronics, Vol. 64, No. 5, pp. 4076-4090, 2016.

[11] S. Y. Wang, W. Chaovalitwongse, and R. Babuska. Machine Learning Algorithms in Bipedal Robot Control. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 42, No. 5, pp. 728-743, 2012.

[12] C. M. Bishop. Pattern Recognition and Machine Learning (Information Science and statistics). Springer-Verlag New York. Inc. Secaucus, NJ, USA, 2006.

[13] J. Shin, T. A. Badgwell, K. H. Liu and J. H. Lee. Reinforcement Learning - Overview of recent progress and implications for process control. Computers and Chemical Engineering, Vol. 127, pp. 282-294, 2019.

[14] R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. MIT Press, 2018.

[15] R. Pinsler, R. Akrour, T. Osa, J. Peters and G. Neumann, Sample and feedback efficient hierarchical reinforcement learning from human preferences. IEEE Int. Conf. Robotics and Automation, 2018.

[16] A. Ferdowsi, U. Challita, W. Saad and N. B. Mandayam, Robust deep reinforcement learning for security and safety in autonomous vehicle systems. Int. Conf. Intelligent Transp. Syst.,, 2018.

[17] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits and Systems Magazine, Vol. 9, No. 3, pp. 32-50, 2009.

[18] X. S. Wang, Y. H. Cheng and W. Sun. A proposal of adaptive PID controller based on reinforcement learning. Journal of China Univ. Mining and Technology, Vol. 17, No. 1, pp. 40-44, 2007.

[19] M. N. Howell and M. C. Best. On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata. Control Engineering Practice, Vol. 8, pp. 147-154, 2000.

[20] Z. S. Jin, H. C. Li and H. M. Gao. An intelligent weld control strategy based on reinforcement learning approach. The International Journal of Advanced Manufacturing Technology, Vol. 100, pp. 2163-2175, 2019.

[21] M. Sedighizadeh and A. Rezazadeh. Adaptive PID Controller based on Reinforcement Learning for Wind Turbine Control. World Academy of Science, Engineering and Technology 13, pp. 267-262, 2008.

[22] A. G. Barto, R. S. Sutton and C. Anderson. Neuron-like adaptive elements that can solve difficult learning control problems. IEEE Trans. Syst., Man, Cybern., Vol. SMC-13, pp. 834-846, 1983.

[23] Suni V. T. Elanayar and Y. C. Shin. Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems. IEEE Transaction on Neural Network, Vol. 5, No. 4, pp. 584-603, 1994.

[24] J. S. Roger Jang and C. T. Sun. Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems. IEEE Transaction on Neural Network, Vol. 4, No. 1, pp. 156-159, 1993.

[25] T. Yamamoto, K. Takao and T. Yamada. Design of a Data-Driven PID controller. IEEE Transaction on Control Systems Technology, Vol. 17, No. 1, pp. 29-39, 2009.

[26] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Netw., Vol. 1, No. 1, pp. 4/27-4/27, 1990.

[27] L. Zi-Qiang. On identification of the controlled plants described by the Hammerstein system. IEEE Trans. Automat. Contr., Vol. Ac-39, No. 2, pp. 569-573, 1994.